# Linux Users and Permissions

## User management:

There are three types of users we create in a Linux OS.

1) root user (administrator) will be only one per machine
2) sudo user (Temporary root user access) to execute commands as root for temporary
3) normal user (Only user specific operations) and System users

### root user configuration

```
root@ubuntudockerserver: ~# pwd

/root

root@ubuntudockerserver: ~#
```

By default, root user ID will be 0 in any Linux OS and IDs 1 to 999 are assigned to the system users. The rest of the users will be assigned from 1000 onwards.

How to check min and max UID range of users: /etc/login.defs (UID_MIN and UID_MAX)

How to list normal users from specified range: #getent passwd {min..max}

User information will be stored in a file /etc/passwd. Sample /etc/passwd file mentioned below for reference


From above output, if I want to list only user's column:

#awk -F':' '{print $1}' /etc/passwd

## sudo user configuration:

By default, Ubuntu installation will not allow root user login. So, while installing Ubuntu we will create a user which will be by default a sudo user as mentioned below.

```
testuser@ubuntudockerserver:~$ pwd
/home/testuser
testuser@ubuntudockerserver:~$ ls -l
total 8
-rw-rw-r-- 1 testuser testuser 4606 Mar 31  2012 sample.war
testuser@ubuntudockerserver:~$
```

To create additional sudo user, we have to create normal user user and then configure/convert him to a sudo user.

Note: That user “testuser” will be added to sudo group by default as part of OS installation.

# How to create users in Linux OS?

There are two commands to create a users: adduser and useradd commands.

adduser command will create user, group, UID, GID, home directory, copies skel file like .bashrc, .profile etc.,, set password.

useradd command it will only creats a user by default, but we have to manually create home directory, set password and copy skel files etc., (we need to pass additional arguments to achieve that)

**adduser command**:

```
root@ubuntudockerserver:~# adduser --help
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] [--add_extra_groups]
[--encrypt-home] USER
   Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]
[--disabled-login] [--add_extra_groups] USER
   Add a system user

adduser --group [--gid ID] GROUP
addgroup [--gid ID] GROUP
   Add a user group

addgroup --system [--gid ID] GROUP
   Add a system group

adduser USER GROUP
   Add an existing user to an existing group

general options:
  --quiet | -q      don't give process information to stdout
  --force-badname   allow usernames which do not match the
                    NAME_REGEX[_SYSTEM] configuration variable
  --extrausers      uses extra users as the database
  --help | -h       usage message
  --version | -v    version number and copyright
  --conf | -c FILE  use FILE as configuration file

root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# adduser
adduser: Only one or two names allowed.
root@ubuntudockerserver:~# adduser sudheer
Adding user `sudheer' ...
Adding new group `sudheer' (1001) ...
Adding new user `sudheer' (1001) with group `sudheer' ...
Creating home directory `/home/sudheer' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sudheer
Enter the new value, or press ENTER for the default
        Full Name []: sudheer reddy duba
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# ls -l /home/sudheer/
total 0
root@ubuntudockerserver:~# ls -al /home/sudheer/
total 20
drwxr-xr-x 2 sudheer sudheer 4096 Mar  6 15:00 .
drwxr-xr-x 5 root    root    4096 Mar  6 15:00 ..
-rw-r--r-- 1 sudheer sudheer  220 Mar  6 15:00 .bash_logout
-rw-r--r-- 1 sudheer sudheer 3771 Mar  6 15:00 .bashrc
-rw-r--r-- 1 sudheer sudheer  807 Mar  6 15:00 .profile
root@ubuntudockerserver:~# cat /etc/passwd | grep sudheer
testuser:x:1000:1000:sudheerdemo:/home/testuser:/bin/bash
sudheer:x:1001:1001:sudheer reddy duba,,,:/home/sudheer:/bin/bash
root@ubuntudockerserver:~# cat /etc/passwd | grep -w sudheer
sudheer:x:1001:1001:sudheer reddy duba,,,:/home/sudheer:/bin/bash
root@ubuntudockerserver:~#
```

From above screenshots, we could see that user was created interactively with all necessary files and folders like home directory, default bash shell, UID, GID etc.,

**useradd command:**

```
root@ubuntudockerserver:~# useradd --help
Usage: useradd [options] LOGIN
       useradd -D
       useradd -D [options]

Options:
      --badnames                do not check for bad names
  -b, --base-dir BASE_DIR       base directory for the home directory of the
                                new account
      --btrfs-subvolume-home    use BTRFS subvolume for home directory
  -c, --comment COMMENT         GECOS field of the new account
  -d, --home-dir HOME_DIR       home directory of the new account
  -D, --defaults                print or change default useradd configuration
  -e, --expiredate EXPIRE_DATE  expiration date of the new account
  -f, --inactive INACTIVE       password inactivity period of the new account
  -g, --gid GROUP               name or ID of the primary group of the new
                                account
  -G, --groups GROUPS           list of supplementary groups of the new
                                account
  -h, --help                    display this help message and exit
  -k, --skel SKEL_DIR           use this alternative skeleton directory
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -l, --no-log-init             do not add the user to the lastlog and
                                faillog databases
  -m, --create-home             create the user's home directory
  -M, --no-create-home          do not create the user's home directory
  -N, --no-user-group           do not create a group with the same name as
                                the user
  -o, --non-unique              allow to create users with duplicate
                                (non-unique) UID
  -p, --password PASSWORD       encrypted password of the new account
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       prefix directory where are located the /etc/* files
  -s, --shell SHELL             login shell of the new account
  -u, --uid UID                 user ID of the new account
  -U, --user-group              create a group with the same name as the user
  -Z, --selinux-user SEUSER     use a specific SEUSER for the SELinux user mapping
      --extrausers              Use the extra users database

root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# useradd teja
root@ubuntudockerserver:~# ls -l /home/teja
ls: cannot access '/home/teja': No such file or directory
root@ubuntudockerserver:~# cat /etc/passwd | grep -w teja
teja:x:1002:1002::/home/teja:/bin/sh
root@ubuntudockerserver:~#
```

Note: we can create duplicate users but UID will be unique

Check the password details of the users as mentioned below

```
root@ubuntudockerserver:~# cat /etc/shadow | grep -w medha
medha:test12345:19057:0:99999:7:::
root@ubuntudockerserver:~# cat /etc/shadow | grep -w teja
teja:!:19057:0:99999:7:::
root@ubuntudockerserver:~# cat /etc/shadow | grep -w sudheer
sudheer:$6$jWKmU3koZmt3RbrD$px4iAsNPmjgxmdUSPApodDpdlWhV4etc7S.j1oqDt1nq5w35Fkno9BiL5OUK5aL0p/lMiQSSTU/4I6U4BoG1J.:19057:0:99999:7:::
root@ubuntudockerserver:~#
```

1. **Username**: It is your login name.
2. **Password**: It is your encrypted password. The password should be minimum 8-12 characters long including special characters, digits, lower case alphabetic and more. Usually, password format is set to `$id$salt$hashed`, the `$id` is the algorithm used on GNU/Linux as follows:
    1. **$1$** is MD5
    2. **$2a$** is Blowfish
    3. **$2y$** is Blowfish
    4. **$5$** is SHA-256
    5. **$6$** is SHA-512
3. **Last password change (last changed)**: Days since Jan 1, 1970, that password was last changed
4. **Minimum**: The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. **Maximum**: The maximum number of days the password is valid (after that user is forced to change his/her password)
6. **Warn**: The number of days before password is to expire that user is warned that his/her password must be changed
7. **Inactive**: The number of days after password expires that account is disabled
8. **Expire**: days since Jan 1, 1970, that account is disabled i.e. an absolute date specifying when the login may no longer be used.

How to create a user with custom password?

#perl -e "print crypt('pass', '\$6\$salt\$')"

#root@ubuntuserverdemo:~# useradd -u 1007 -m -d /home/teja -s /bin/bash -c "teja user was created for testing" --password $(perl -e "print crypt('test123456', '\$6\$salt\$')") teja

Or

#root@ubuntuserverdemo:~# echo sudheerdemo:test123456 | chpasswd

Some of important files:

cat /etc/login.defs

cat /etc/default/useradd

cat /etc/passwd

cat /etc/shadow

cat /etc/group

```
root@ubuntudemoserveraws:~# cat /etc/sudoers.d/visualpath
visualpath ALL=(ALL:ALL) NOPASSWD:/bin/rmdir,/bin/mkdir
root@ubuntudemoserveraws:~#
```

```
root@ubuntudemoserveraws:~# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
root@ubuntudemoserveraws:~#
```

- root ALL=(ALL:ALL) ALL The first field indicates the username that the rule will apply to (**root**).
- root ALL=(ALL:ALL) ALL The first “ALL” indicates that this rule applies to all hosts.
- root ALL=(ALL:ALL) ALL This “ALL” indicates that the **root** user can run commands as all users.
- root ALL=(ALL:ALL) ALL This “ALL” indicates that the **root** user can run commands as all groups.
- root ALL=(ALL:ALL) ALL The last “ALL” indicates these rules apply to all commands.

```
root@ubuntudemoserveraws:~# usermod -aG sudo visualpath
root@ubuntudemoserveraws:~#
root@ubuntudemoserveraws:~# cat /etc/group | grep sudo
sudo:x:27:testuser,visualpath
root@ubuntudemoserveraws:~#
```

How to add a normal user to sudo group to ensure user can be sudo user as mentioned below

```
root@ubuntudockerserver:~# usermod -aG sudo sudheer
root@ubuntudockerserver:~# su - sudheer
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

sudheer@ubuntudockerserver:~$ pwd
/home/sudheer
sudheer@ubuntudockerserver:~$ sudo apt update
[sudo] password for sudheer:
Hit:1 https://download.docker.com/linux/ubuntu focal InRelease
Hit:2 https://deb.nodesource.com/node_16.x focal InRelease
Hit:3 https://packages.microsoft.com/repos/azure-cli focal InRelease
Hit:4 http://archive.ubuntu.com/ubuntu focal InRelease
Get:5 http://archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:7 http://archive.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Fetched 336 kB in 5s (74.4 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
56 packages can be upgraded. Run 'apt list --upgradable' to see them.
sudheer@ubuntudockerserver:~$
```

Users deletion

```
root@ubuntudockerserver:~# ls -l /home/medha/
total 0
root@ubuntudockerserver:~# deluser medha
Removing user `medha' ...
Warning: group `medha' has no more members.
Done.
root@ubuntudockerserver:~# ls -l /home/medha/
total 0
root@ubuntudockerserver:~# ls -l /home/
total 16
drwxr-xr-x 3 root     root     4096 Feb 16 11:24 linuxbrew
drwxr-xr-x 2     1005     1005 4096 Mar  6 15:10 medha
drwxr-xr-x 2 sudheer  sudheer  4096 Mar  6 16:20 sudheer
drwxr-xr-x 3 testuser testuser 4096 Mar  6 12:33 testuser
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# ls -l /home/sudheer/
total 0
root@ubuntudockerserver:~# deluser --remove-home sudheer
Looking for files to backup/remove ...
Removing files ...
Removing user `sudheer' ...
Warning: group `sudheer' has no more members.
userdel: user sudheer is currently used by process 9215
/usr/sbin/deluser: `/sbin/userdel sudheer' returned error code 8. Exiting.
root@ubuntudockerserver:~# ls -l /home/sudheer/
ls: cannot access '/home/sudheer/': No such file or directory
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# userdel sudheer
root@ubuntudockerserver:~# getent group sudheer
root@ubuntudockerserver:~# cat /etc/passwd | grep sudheer
testuser:x:1000:1000:sudheerdemo:/home/testuser:/bin/bash
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# chage -l krishna
Last password change                                    : Mar 07, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@ubuntudockerserver:~#
```

How to delete a user along with home directory and mail?

#userdel [-f] [-r] username

How to change user ID?

#usermod -u new_id username

How to change group ID?

#usermod -g new_group_id username

How to change user login name?

#usermod -l new_name old_name

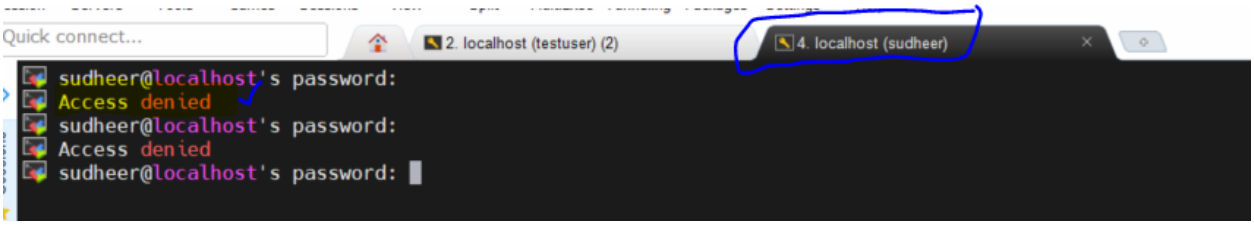
How to change a home directory to new location?

#usermod -d new_home_directory username

**Note: Settings of useradd command are stored in the "/etc/defaults/useradd" file**

How to lock a user?

```
root@ubuntudockerserver:~# adduser sudheer
Adding user `sudheer' ...
Adding new group `sudheer' (1001) ...
Adding new user `sudheer' (1001) with group `sudheer' ...
Creating home directory `/home/sudheer' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sudheer
Enter the new value, or press ENTER for the default
        Full Name []: sudheer reddy duba
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@ubuntudockerserver:~# su - sudheer
sudheer@ubuntudockerserver:~$ pwd
/home/sudheer
sudheer@ubuntudockerserver:~$ ls -al
total 20
drwxr-xr-x 2 sudheer sudheer 4096 Mar  7 02:04 .
drwxr-xr-x 6 root    root    4096 Mar  7 02:04 ..
-rw-r--r-- 1 sudheer sudheer  220 Mar  7 02:04 .bash_logout
-rw-r--r-- 1 sudheer sudheer 3771 Mar  7 02:04 .bashrc
-rw-r--r-- 1 sudheer sudheer  807 Mar  7 02:04 .profile
sudheer@ubuntudockerserver:~$ exit
logout
root@ubuntudockerserver:~# usermod -L sudheer
root@ubuntudockerserver:~# su - sudheer
sudheer@ubuntudockerserver:~$ touch test1
sudheer@ubuntudockerserver:~$ ls -l
total 0
-rw-rw-r-- 1 sudheer sudheer 0 Mar  7 02:05 test1
sudheer@ubuntudockerserver:~$
```




```
root@ubuntudockerserver:~# usermod -U sudheer
root@ubuntudockerserver:~# chage -l sudheer
Last password change                                    : Mar 07, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# cat /etc/passwd | grep -w sudheer
sudheer:x:1001:1001:sudheer reddy duba,,,:/home/sudheer:/bin/bash
root@ubuntudockerserver:~# cat /etc/shadow | grep -w sudheer
sudheer:$6$YQPCbROPhiorxV.l$olx92TFPrgpor4mLBxHvlQho8INskDXzNAUmgwUbLD1trCw.tf9C0FN4iXYHAFlPTnvZEelSBNyr8wgMDV.aO/:19058:0:99999:7:::
root@ubuntudockerserver:~#
```

```
root@ubuntudockerserver:~# passwd -l sudheer
passwd: password expiry information changed.
root@ubuntudockerserver:~# chage -l sudheer
Last password change                                    : Mar 07, 2022
Password expires                                        : never
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 99999
Number of days of warning before password expires       : 7
root@ubuntudockerserver:~# cat /etc/shadow | grep -w sudheer
sudheer:!$6$YQPCbROPhiorxV.l$olx92TFPrgpor4mLBxHvlQho8INskDXzNAUmgwUbLD1trCw.tf9C0FN4iXYHAFlPTnvZEelSBNyr8wgMDV.aO/:19058:0:99999:7:::
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# passwd -L sudheer
passwd: invalid option -- 'L'
Usage: passwd [options] [LOGIN]

Options:
  -a, --all                     report password status on all accounts
  -d, --delete                  delete the password for the named account
  -e, --expire                  force expire the password for the named account
  -h, --help                    display this help message and exit
  -k, --keep-tokens             change password only if expired
  -i, --inactive INACTIVE       set password inactive after expiration
                                to INACTIVE
  -l, --lock                    lock the password of the named account
  -n, --mindays MIN_DAYS        set minimum number of days before password
                                change to MIN_DAYS
  -q, --quiet                   quiet mode
  -r, --repository REPOSITORY   change password in REPOSITORY repository
  -R, --root CHROOT_DIR         directory to chroot into
  -S, --status                  report password status on the named account
  -u, --unlock                  unlock the password of the named account
  -w, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
  -x, --maxdays MAX_DAYS        set maximum number of days before password
                                change to MAX_DAYS

root@ubuntudockerserver:~# passwd -u sudheer
passwd: password expiry information changed.
root@ubuntudockerserver:~# cat /etc/shadow | grep -w sudheer
sudheer:$6$YQPCbROPhiorxV.l$olx92TFPrgpor4mLBxHvlQho8INskDXzNAUmgwUbLD1trCw.tf9C0FN4iXYHAFlPTnvZEelSBNyr8wgMDV.aO/:19058:0:99999:7:::
root@ubuntudockerserver:~#
```

Below commands will provide the status of user password has been locked or note L stands for lock

```
root@ubuntudockerserver:~# passwd -S sudheer
sudheer P 03/07/2022 0 99999 7 -1
root@ubuntudockerserver:~# passwd -l sudheer
passwd: password expiry information changed.
root@ubuntudockerserver:~# passwd -S sudheer
sudheer L 03/07/2022 0 99999 7 -1
root@ubuntudockerserver:~# passwd -u sudheer
passwd: password expiry information changed.
root@ubuntudockerserver:~# passwd -S sudheer
sudheer P 03/07/2022 0 99999 7 -1
root@ubuntudockerserver:~#
```

## Groups configuration:

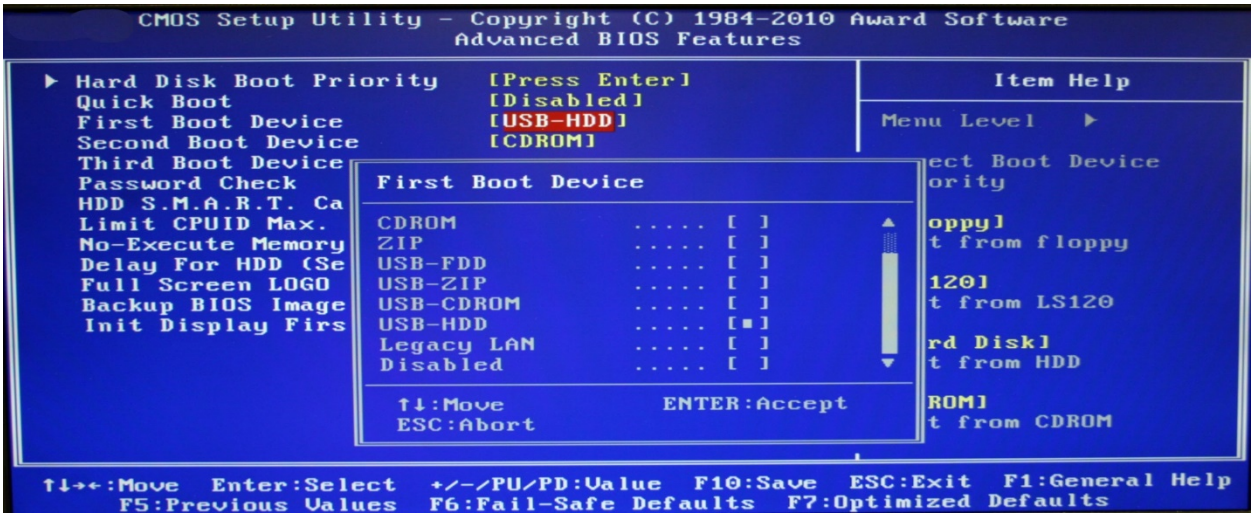
```
root@ubuntudockerserver:~# groups sudheer
sudheer : sudheer
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# groupadd visualpath
root@ubuntudockerserver:~# cat /etc/group | grep -w visualpath
visualpath:x:1003:
root@ubuntudockerserver:~# cat /etc/group | grep visualpath
visualpath:x:1003:
root@ubuntudockerserver:~#
root@ubuntudockerserver:~# groupadd -aG visualpath sudheer
groupadd: invalid option -- 'a'
Usage: groupadd [options] GROUP

Options:
  -f, --force                   exit successfully if the group already exists,
                                and cancel -g if the GID is already used
  -g, --gid GID                 use GID for the new group
  -h, --help                    display this help message and exit
  -K, --key KEY=VALUE           override /etc/login.defs defaults
  -o, --non-unique              allow to create groups with duplicate
                                (non-unique) GID
  -p, --password PASSWORD       use this encrypted password for the new group
  -r, --system                  create a system account
  -R, --root CHROOT_DIR         directory to chroot into
  -P, --prefix PREFIX_DIR       directory prefix
      --extrausers              Use the extra users database

root@ubuntudockerserver:~# usermod -aG visualpath sudheer
root@ubuntudockerserver:~# groups sudheer
sudheer : sudheer visualpath
root@ubuntudockerserver:~#
```

# Disk and Partitions

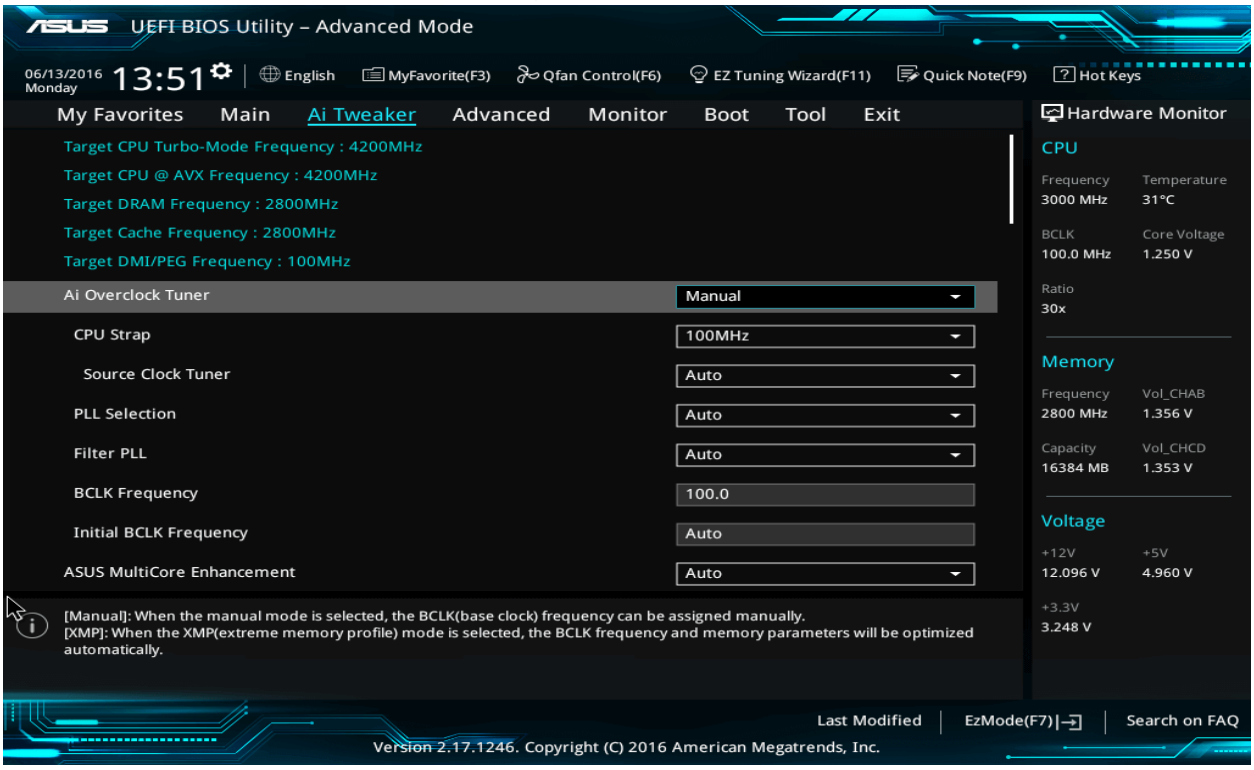
## Disk Partition:

1) It helps to separate physical disk into logical pieces.
2) It's not mandatory to create partitions on a disk.
3) We can create separate partition to store Operating System, in an event of crash. So that application data is not lost.

**BIOS (Basic Input Output System)**



**UEFI (Unified Extensible Firmware Interface)**




# Partitions

A hard disk can be divided into several *partitions*. Each partition functions as if it were a separate hard disk. The idea is that if you have one hard disk, and want to have, say, two operating systems on it, you can divide the disk into two partitions. Each operating system uses its partition as it wishes and doesn't touch the other ones. This way the two operating systems can co-exist peacefully on the same hard disk. Without partitions one would have to buy a hard disk for each operating system.

Floppies are not usually partitioned. There is no technical reason against this, but since they're so small, partitions would be useful only very rarely. CD-ROMs are usually also not partitioned, since it's easier to use them as one big disk, and there is seldom a need to have several operating systems on one.

1. The MBR, boot sectors and partition table

The information about how a hard disk has been partitioned is stored in its first sector (that is, the first sector of the first track on the first disk surface). The first sector is the *master boot record* (MBR) of the disk; this is the sector that the BIOS reads in and starts when the machine is first booted. The master boot record contains a small program that reads the partition table, checks which partition is active (that is, marked bootable), and reads the first sector of that partition, the partition's *boot sector* (the MBR is also a boot sector, but it has a special status

and therefore a special name). This boot sector contains another small program that reads the first part of the operating system stored on that partition (assuming it is bootable), and then starts it.

The partitioning scheme is not built into the hardware, or even into the BIOS. It is only a convention that many operating systems follow. Not all operating systems do follow it, but they are the exceptions. Some operating systems support partitions, but they occupy one partition on the hard disk, and use their internal partitioning method within that partition. The latter type exists peacefully with other operating systems (including Linux), and does not require any special measures, but an operating system that doesn't support partitions cannot co-exist on the same disk with any other operating system.

As a safety precaution, it is a good idea to write down the partition table on a piece of paper, so that if it ever corrupts you don't have to lose all your files. (A bad partition table can be fixed with **fdisk**). The relevant information is given by the **fdisk -l** command:

```
$ fdisk -l /dev/sda

Disk /dev/sda: 15 heads, 57 sectors, 790 cylinders
Units = cylinders of 855 * 512 bytes

   Device Boot  Begin   Start     End  Blocks   Id  System
/dev/sda1         1       1      24   10231+ 82  Linux swap
/dev/sda2        25      25      48   10260   83 Linux native
/dev/sda3        49      49     408  153900   83  Linux native
/dev/sda4       409     409     790  163305    5  Extended
/dev/sda5       409     409     744  143611+  83  Linux native
/dev/sda6       745     745     790   19636+  83  Linux native
$
```

2. Extended and logical partitions

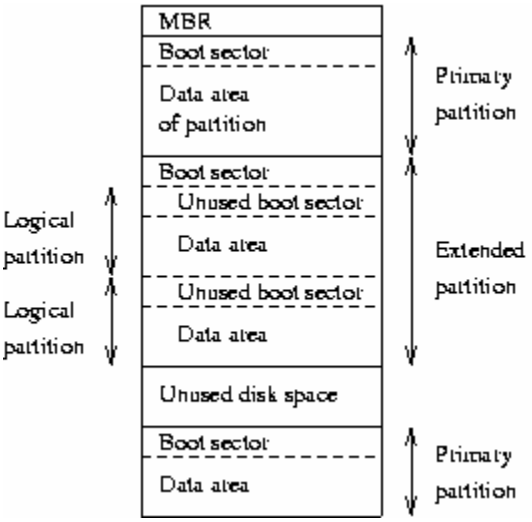
The original partitioning scheme for PC hard disks allowed only four partitions. This quickly turned out to be too little in real life, partly because some people want more than four operating systems (Linux, MS-DOS, OS/2, Minix, FreeBSD, NetBSD, or Windows/NT, to name a few), but primarily because sometimes it is a good idea to have several partitions for one operating system. For example, swap space is usually best put in its own partition for Linux instead of in the main Linux partition for reasons of speed (see below).

To overcome this design problem, *extended partitions* were invented. This trick allows partitioning a *primary partition* into sub-partitions. The primary partition thus subdivided is the *extended partition*; the sub-partitions are *logical partitions*. They behave like primary

partitions but are created differently. There is no speed difference between them. By using an extended partition, you can now have up to 15 partitions per disk.

The partition structure of a hard disk might look like that in Figure 5-2. The disk is divided into three primary partitions, the second of which is divided into two logical partitions. Part of the disk is not partitioned at all. The disk as a whole and each primary partition has a boot sector.

**Figure 5-2. A sample hard disk partitioning.**

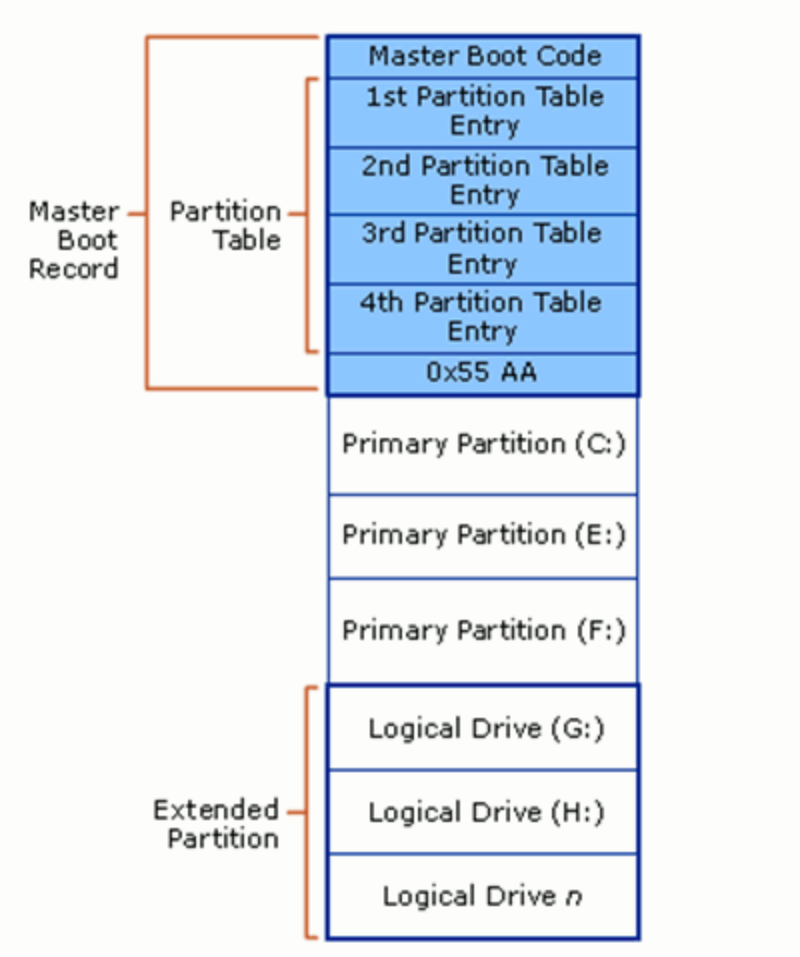



**Master Boot Recorder:**

MBR is the old standard for managing the partition in the hard disk, and it is still being used extensively by many people. The MBR resides at the very beginning of the hard disk, and it holds the information on how the logical partitions are organized in the storage device. In addition, the MBR also contains executable code that can scan the partitions for the active OS and load up the boot up code/procedure for the OS.

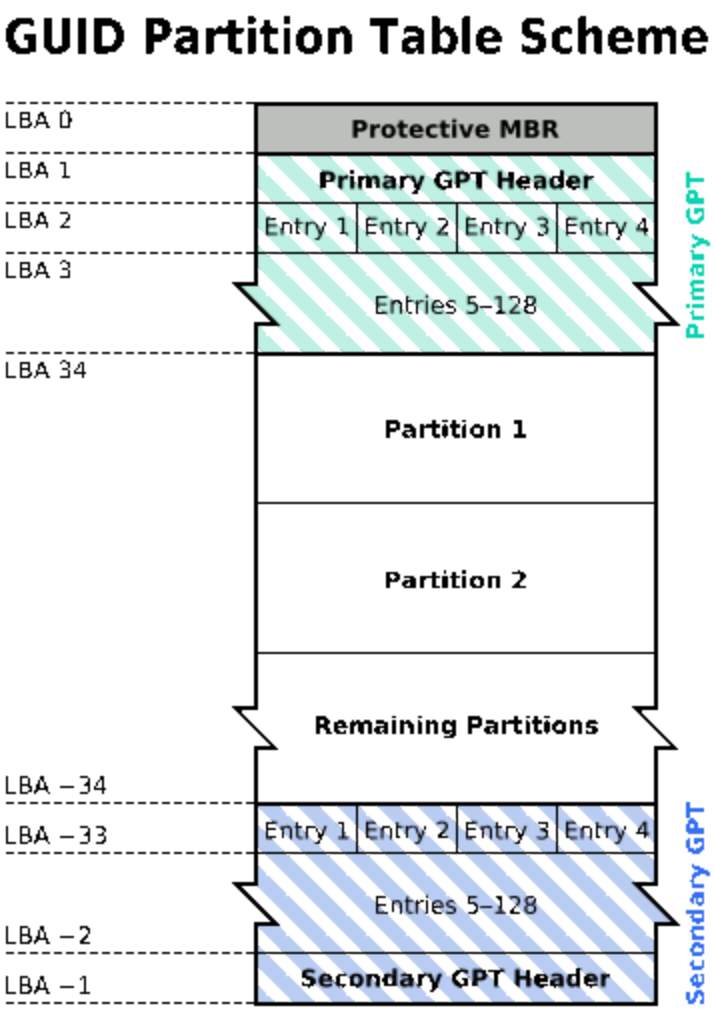
For a MBR disk, you can only have four primary partitions. To create more partitions, you can set the fourth partition as the extended partition, and you will be able to create more sub-partitions (or logical drives) within it. As MBR uses 32-bit to record the partition, each partition can only go up to a maximum of 2TB in size.

There are several pitfalls with MBR. First, you can only have 4 partitions in the hard disk and each partition is limited to only 2TB in size. This is not going to work well with hard disk of big storage space, say 100TB. Secondly, the MBR is the only place that holds the partition information. If it ever gets corrupted (and yes, it can get corrupted very easily), the entire hard disk is unreadable.




## GUID Partition Table (GPT)

GPT is the latest standard for laying out the partitions of a hard disk. It makes use of globally unique identifiers (GUID) to define the partition and it is part of the UEFI standard. This means that on a UEFI-based system (which is required for Windows 8 Secure Boot feature), it is a must to use GPT. With GPT, you can create theoretically unlimited partitions on the hard disk, even though it is generally restricted to 128 partitions by most OSes. Unlike MBR that limits each partition to only 2TB in size, each partition in GPT can hold up to 2^64 blocks in length (as it is using 64-bit), which is equivalent to 9.44ZB for a 512-byte block (1 ZB is 1 billion terabytes). In Microsoft Windows, that size is limited to 256TB.




From the GPT Table Scheme diagram above, you can see that there is a primary GPT at the beginning of the hard disk and a secondary GPT at the end. This is what makes GPT more useful than MBR. GPT stores a backup header and partition table at the end of the disk so it can be recovered if the primary tables are corrupted. It also carries out CRC32 checksums to detect errors and corruption of the header and partition table.

You can also see that there is a protective MBR at the first sector of the hard disk. Such hybrid setup is to allow a BIOS-based system to boot from a GPT disk using a boot loader stored in the protective MBR's code area. In addition, it protects the GPT disk from damage by GPT-unaware disk utilities.

3. Partition types

The partition tables (the one in the MBR, and the ones for extended partitions) contain one byte per partition that identifies the type of that partition. This attempts to identify the operating system that uses the partition, or what it uses it for. The purpose is to make it possible to avoid having two operating systems accidentally using the same partition. However operating systems do not really care about the partition type byte, e.g., Linux doesn't care at all what it is. Worse, some of them use it incorrectly, e.g., at least some versions of DR-DOS ignore the most significant bit of the byte, while others don't.

There is no standardization agency to specify what each byte value means, but as far as Linux is concerned, here is a list of partition types as per the **fdisk** program.

```
 0  Empty           1c  Hidden Win95 FA 70  DiskSecure Mult bb  Boot Wizard hid
 1  FAT12           1e  Hidden Win95 FA 75  PC/IX           be  Solaris boot
 2  XENIX root      24  NEC DOS         80  Old Minix       c1  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          81  Minix / old Lin c4  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic  82  Linux swap      c6  DRDOS/sec (FAT-
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 6  FAT16           41  PPC PReP Boot   84  OS/2 hidden C:  da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux extended  db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  NTFS volume set de  Dell Utility
 9  AIX bootable    4e  QNX4.x 2nd part 87  NTFS volume set df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 b  Win95 FAT32     50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  Win95 FAT32 (LB 51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 e  Win95 FAT16 (LB 52  CP/M           9f  BSD/OS          eb  BeOS fs
 f  Win95 Ext'd (LB 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
10  OPUS           54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
12  Compaq diagnost 56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
14  Hidden FAT16 <3 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor       a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Darwin boot     fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden Win95 FA 65  Novell Netware  b8  BSDI swap       ff  BBT
```

4. Partitioning a hard disk

There are many programs for creating and removing partitions. Most operating systems have their own, and it can be a good idea to use each operating system's own, just in case it does something unusual that the others can't. Many of the programs are called **fdisk**, including the Linux one, or variations thereof. Details on using the Linux **fdisk** given on its man page.
The **cfdisk** command is like **fdisk** but has a nicer (full screen) user interface.

When using IDE disks, the boot partition (the partition with the bootable kernel image files) must be completely within the first 1024 cylinders. This is because the disk is used via the BIOS during boot (before the system goes into protected mode), and BIOS can't handle more than 1024 cylinders. It is sometimes possible to use a boot partition that is only partly within the first 1024 cylinders. This works if all the files that are read with the BIOS are within the first 1024 cylinders. Since this is difficult to arrange, it is *a very bad idea* to do it; you never know when a kernel update or disk defragmentation will result in an unbootable system. Therefore, make sure your boot partition is completely within the first 1024 cylinders.

However, this may no longer be true with newer versions of LILO that support LBA (Logical Block Addressing). Consult the documentation for your distribution to see if it has a version of LILO where LBA is supported.

Some newer versions of the BIOS and IDE disks can, in fact, handle disks with more than 1024 cylinders. If you have such a system, you can forget about the problem; if you aren't quite sure of it, put it within the first 1024 cylinders.

Each partition should have an even number of sectors, since the Linux filesystems use a 1 kilobyte block size, i.e., two sectors. An odd number of sectors will result in the last sector being unused. This won't result in any problems, but it is ugly, and some versions of **fdisk** will warn about it.

Changing a partition's size usually requires first backing up everything you want to save from that partition (preferably the whole disk, just in case), deleting the partition, creating new partition, then restoring everything to the new partition. If the partition is growing, you may need to adjust the sizes (and backup and restore) of the adjoining partitions as well.

Since changing partition sizes is painful, it is preferable to get the partitions right the first time or have an effective and easy to use backup system. If you're installing from a media that does not require much human intervention (say, from CD-ROM, as opposed to floppies), it is often easy to play with different configuration at first. Since you don't already have data to back up, it is not so painful to modify partition sizes several times.

There is a program for MS-DOS, called **fips**, which resizes an MS-DOS partition without requiring the backup and restore, but for other filesystems it is still necessary.

The **fips** program is included in most Linux distributions. The commercial partition manager ``Partition Magic'' also has a similar facility but with a nicer interface. Please do remember that partitioning is dangerous. Make *sure* you have a recent backup of any important data before you try changing partition sizes ``on the fly''. The program **parted** can resize other types of partitions as well as MS-DOS, but sometimes in a limited manner. Consult the **parted** documentation before using it, better safe than sorry.

5. Device files and partitions

Each partition and extended partition have its own device file. The naming convention for these files is that a partition's number is appended after the name of the whole disk, with the convention that 1-4 are primary partitions (regardless of how many primary partitions there are) and number greater than 5 are logical partitions (regardless of within which primary partition they reside). For example, /dev/hda1 is the first primary partition on the first IDE hard disk, and /dev/sdb7 is the third extended partition on the second SCSI hard disk.

# Linux Logical Volume Manager

**Creating a Physical Volume**

#pvcreate /dev/sdb /dev/sdc /dev/sdd

#pvdisplay

#pvscan

#pvs

**Create a Volume Group**

#vgcreate docker-data /dev/sdb /dev/sdc

#pvscan

#vgscan

#vgdisplay docker-data

#vgextend docker-data /dev/sdd

#vgs

**Creating a Logical Volumes**

Linear, Mirror and Stripe

**Linear**: PVs are filled up sequentially rather than in parallel

**Mirrored**: If one mirror were to fail the good mirror would still be available enable the file system or swap device contained with in to remain in operation

**Striped**: A strip unit defines how much data is written to each PV in one go and writes are alternated between the PVs

#lvcreate -L 5G -nlvdata docker-data

#lvdisplay

#lvscan

#lvs

#pvdisplay -m /dev/sdb

#lvextend -l +100%FREE /dev/docker-data/lvdata

How to perform filesystem check

#e2fsck -f /dev/docker-data/lvdata

**Increate parition**

#lvextend -L +1G /dev/docker-data/lvdata

#resize2fs /dev/docker-data/lvdata

#mount

#df -h

**Decrease parition**

#umount -v[verbose] /mnt

#e2fsck -ff /dev/docker-data/lvdata

#resize2fs /dev/docker-data/lvdata 2G  [pass a value based on what should be size at last, not how much you want to reduce Example: reduce from 10GB → 8G, so we have to pass 2G]

#lvreduce -L 8G /dev/docker-data/lvdata

Note: we have to pass the value that should match with the size of the that we are looking for atlast.

#resize2fs /dev/docker-data/lvdata

#mount /dev/docker-data/lvdata /mnt

#df -h

root@ip-172-31-30-28:~# pvscan

  PV /dev/xvdf   VG sudheer-demo    lvm2 [<50.00 GiB / 0    free]

  PV /dev/xvdg   VG sudheer-demo    lvm2 [<100.00 GiB / 0    free]

  PV /dev/xvdh   VG sudheer-demo    lvm2 [<50.00 GiB / 0    free]

  PV /dev/xvdi   VG sudheer-demo    lvm2 [<100.00 GiB / 0    free]

  Total: 4 [299.98 GiB] / in use: 4 [299.98 GiB] / in no VG: 0 [0   ]

root@ip-172-31-30-28:~# vgscan

  Found volume group "sudheer-demo" using metadata type lvm2

root@ip-172-31-30-28:~# lvscan

  ACTIVE            '/dev/sudheer-demo/docker-data' [299.98 GiB] inherit

root@ip-172-31-30-28:~#

Boot information:

| Runlevel | Target |
|---|---|
| 0 | Poweroff.target |
| 1 | Rescue.target |
| 2, 3, 4 | Multi-user.target |
| 5 | Graphical.target |
| 6 | Reboot.target |

1) How to list status of all service that start/do not start as part of booting your linux system?
    a. systemctl list-units --type service
2) List only active services
    a. systemctl list-units --type service --state active
    b. systemctl -t service --state active
3) List all enabled service (all active and non-active services)
    a. systemctl list-unit-files --type service
    b. systemctl list-unit-files --state enabled
    c. systemctl list-unit-files --type enabled --state enabled
4) List services that start before actual service
    a. systemctl list-dependencies --after sshd
5) start, stop, restart a services
    a. systemctl status <service_name>. service
    b. systemctl start <service_name>. service
    c. systemctl stop <service_name>. service
    d. systemctl reload <service_name>. service
    e. systemctl enable <service_name>. service
    f. systemctl disable <service_name>.service
    g. systemctl mask <service_name>. service
    h. systemctl unmask <service_name>. service
    i. systemctl try-restart <service_name>. service
    j. systemctl is-active <service_name>. service
    k. systemctl is-enabled <service_name>. service
    Note: service_name replace with actual service name to check like sshd, docker etc.,
6) How to find or read errors while managing services?
    a. journalctl -u <service_name>
    b. journalctl -xe

**Permissions and Ownership for Files and Directories**:

Demo users created using adduser command:
root@ip-172-31-22-173:~# ls -l /home/
total 16
drwxr-x--- 2 sudhams sudhams 4096 Jun  1 14:12 sudhams
drwxr-x--- 2 sudheer sudheer 4096 Jun  1 14:11 sudheer
drwxr-x--- 2 teja    teja    4096 Jun  1 14:11 teja
drwxr-x--- 4 ubuntu  ubuntu  4096 Jun  1 14:11 ubuntu

root@ip-172-31-22-173:~#

r → read permission to read the content of the file, for directory it is list files under the directory if x is also set
w → write or modify the file content, for directory to create, delete, rename/update files if x is also set
x → execute permission on file, like program or script to execute. For directory it is allowed to enter using cd command
- → Means permissions are not assigned

| Binary | Decimal | Octet | permission |
|---|---|---|---|
| 000 | 000 | 0 | --- |
| 001 | 001 | 1 | r-- |
| 010 | 020 | 2 | -w- |
| 011 | 021 | 3 | -wx |
| 100 | 400 | 4 | r-- |
| 101 | 401 | 5 | r-x |
| 110 | 420 | 6 | rw- |
| 111 | 421 | 7 | rwx |

User → permission will be applied only to owner of the file or directory, it will not affect other users
Group → permission will be applied to the group owner of the file or directory, it will not affect other users who are not part of that group
Others → permission will be applied to all other users on the system, we must be careful in changing these permissions

Commands:
#chmod permissions file/directory and chgrp to change the group owner of a file/directory
+ → to add permissions
- → to remove permissions
= → to assign permissions

#chmod +x file → add execute permissions to all users, groups, and others
#chmod -x file → to remove execute permissions to all
#chmod u+x file
#chmod u-x file
#chmod ug+rw file
#chmod ugo-rw file
#chgrp new_group file → change group name to new one, but this can be done by owner of the file/directory